10-01-04

AF $ 120

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In Re Application of:

Dennis L. Montgomery

Serial No. **09/727,096**
Filed: **November 29, 2000**

For: **Method and Apparatus for Encoding Information Using Multiple Passes and Decoding In A Single Pass**

Examiner: **H. Mahmoudi**
Group Art Unit: **2175**

Atty Docket No. 042503-0261928
ET-001

Confirmation No. 6817

APPELLANT'S BRIEF ON APPEAL
(37 C.F.R. 1.192)

BOARD OF PATENT APPEALS & INTERFERENCES

Mail Stop: Appeal Briefs-Patents
Commissioner of Patents
P.O. Box 1450
Alexandria, VA 22313

This brief is transmitted in triplicate. (37 C.F.R. 1.192(a))

Express Mail Label No. EV 382 008 390 US
Date of Deposit: September 29, 2004

I hereby certify that this paper and attachments identified below are being deposited with the United States Postal Service as "Express Mail Post Office to Addressee, under 35 CFR 1.10 addressed to Mail Stop Appeal Briefs-Patents, Commissioner of Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date indicated above.

By: Kathleen M Smith
Kathleen M. Smith

APPEAL BRIEF FEE

The Commissioner is authorized to charge Deposit Account 50-2213 (Order No. 042503-0261928) for the small entity fee of $165.00 for filing a brief in support of appeal, and for any other fees associated with this appeal.

REQUEST FOR EXTENSION OF TIME

This is a Brief on Appeal, following the Notice of Appeal filed March 29, 2004. Appellants respectfully request a four-month extension of time, from May 29, 2004 to the date of this submission in order to file the accompanying Brief on Appeal. The Commissioner is authorized to charge the requisite small-entity extension fee of ***740.00*** to Deposit Account 50-2213 (Order No. 042503-0261928).

HEADINGS

This brief contains these items under the following headings and in the order set forth below (37 C.F.R. 1.192(c)):

I. REAL PARTY IN INTEREST
II. RELATED APPEALS AND INTERFERENCES
III. STATUS OF CLAIMS
IV. STATUS OF AMENDMENTS
V. SUMMARY OF INVENTION
VI. ISSUES
VII. GROUPING OF CLAIMS
VIII. ARGUMENTS
IX. APPENDIX OF CLAIMS ON APPEAL
X. OTHER MATERIALS THAT APPELLANT CONSIDERS NECESSARY OR DESIRABLE

I. REAL PARTY IN INTEREST

(37 C.F.R. 1.192(c)(1))

The real party in interest in this appeal is **eTtreppid Technologies, LLC.**

II. RELATED APPEALS AND INTERFERENCES

(37 C.F.R. 1.192(c)(2))

There are no other appeals or interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

(37 C.F.R. 1.192(c)(3))

Claims 1-73 are pending in the above referenced application, of which the Examiner has indicated that claims 22-28 and 30-38 contain allowable subject matter and claims 1-21, 29, and 39-73 were finally rejected in the Office Action mailed September 25, 2003.

IV. STATUS OF AMENDMENTS

(37 C.F.R. 1.192(c)(4))

A Reply to Office Action (the "Reply") under Rule 116 was filed on March 25, 2004. It was filed and considered, but not deemed to place the application in condition for allowance. *See* Advisory Action mailed April 20, 2004. No further submissions have been made since the filing of the Reply.

V. SUMMARY OF INVENTION

(37 C.F.R. 1.192(c)(5))

The present invention discloses a method and system that include various patentably distinct embodiments, such as compression of multiple threads simultaneously, multiple passes of compression of threads, and compression in multiple passes with decompression in a single pass. The discussion below describes as required the inventions with reference to the specification, which

is the Substitute Specification filed on April 12, 2001. The references are not intended to limit the scope of the recited inventions.

The compression and decompression techniques disclosed in the present invention are not limited by the type of files that can be operated upon by the system shown in Fig. 2, which is attached hereto. The system, such as system 200, includes a controller interface 220 and the compression/decompression (C/D) engine 230. The C/D engine can be implemented using at least one processor that has the capability of operating upon multiple threads simultaneously. *See specification, page 6, lines 10-22.* With respect to the compression of data using multiple threads, the C/D engine performs the compression of the data. *Id.* Additionally, multiple processors can be used in parallel to even more efficiently implement the C/D engine. *Id. at page 6, line 18.* Prior to the compression operation, various grouping methods can take place. For example, the data portions of the threads are grouped together based on some relative similarity. *Id. at page 8, line 16.* The grouping of the data can be based on file type, subject matter for similar files. *Id. at page 9, lines 6-10.* Since there are many different types of image files, program files, and compressed files, each type of file (as identified in the header of each file) is grouped relative to each other. *Id.* There could, alternatively be subject matter groupings for similar file types, such as the image, program and compressed files. Alternatively, grouping could be determined by the adaptively predicted time that will be needed for compression, in which case group 1 files are those that are predicted to compress the fastest, whereas the group N files are those that are predicted to compress the slowest. *Id. at lines 16-19.*

Once the grouping process is completed, the interface controller, which is labeled element 220 in Figure 2, operates upon each block to adaptively predict the period of time that will be required to compress each of the blocks for each of the files. *Id. at page 10, lines 3-10.* All of the blocks of a particular file type are estimated as being the same for the same relative block size based upon the file type from the header. In this regard, the estimated amount of compression and estimated time it will take to achieve that compression using a specific compression routine for each file type can be maintained in a table and the amount of compression will generally increase as a greater number of passes are used. *Id. at lines 12-15.* For example, during the compression process, the estimates of the amount of compression needed and the number of passes to get to that

level of compression, and estimated time required to achieve that level of compression can be obtained. *Id. at page 11, lines 13-22.*

The compression process includes selection of the appropriate compression routine based on control signal and metadata related to the thread being compressed. *Id. at page 14, line 11-13.* With respect to the creation of control signals and metadata, another aspect of the present invention relating to the concept of using different threads for the compression of different data. The metadata that is generated provides characteristics of the compression routine, as well as significant patterns that may be associated with the type of stream that is being operated upon. A different thread may be determined as being needed for each block in a file, or a number of files all may use the same thread. *Id. at page 12, line 11-22.* The interface controller will estimate the number of encoding passes that the routine may make on each block, for that compression routine. Accordingly, if there is a certain block that interface controller has predicted will be difficult to compress, then a separate thread can be identified, with that thread having associated with it unique metadata as well as control signals, that provide the information necessary for the C/D engine to begin the compression routine operation on that thread.

As indicated above, compression of different data using multiple threads is only one aspect of the present invention. The ability to perform multiple compressions on the same data using the metadata is yet another aspect of the present invention. For example, during the compression process, the data is operated upon such that a compression algorithm repeatedly, in a cyclical manner, compresses data that in a previous pass was already compressed by the compression engine. *Id. at page 3, line 20.* Between each of the compression passes, the then compressed data is operated upon using metadata established in the previous pass to eliminate redundancies that exist therein. *Id. at page 3, line 22.* The present invention can operate independently on each block. *Id. at page 12, line 23.* Thus, the interface controller needs to be able to determine when to create a new thread or when to use the same thread for multiple blocks. Once the thread determination is complete, the appropriate control signals, metadata, and blocks of data are transmitted to the C/D engine 230, so that the compression of each of the blocks within a give thread can take place and the first pass of compression is initiated. *Id. at page 14, lines 11-13.*

After the result of the first pass of the compression routine on the initial block or blocks the results, which may be interim, are obtained, and stored in buffer manager. *Id. at page 16, line 5.* After completion of the first pass compression by the C/D engine and removal of the threads associated with the first pass compression, the compressed data is associated with second threads, and thereafter the compressed data is operated upon at the initiation of the second pass using the seconds threads. Once further compression using the second threads are completed, the second threads are removed, preserving the second blocks that have then been twice compressed. *Id. at page 16, line 10; page 17, line 13; page 19, line 6.*

The type of comparison operation that is used can be changed between passes, in an adaptive manner. More specifically, the adaptive determination of the comparison operation is made based upon the pattern of the compressed blocks as compared to representative file type patterns, which representative file type patterns can also be stored in a table on the system, as has been described above.

It is also noted that in the second or subsequent pass of compression, that the metadata used can correspond to metadata created for each block within the thread of a previous pass. *Id at page 20, line 17.* Metadata obtained from one compression operation, and particularly metadata that exists after three, four or five passes have been made, can be saved in that state, and then used as metadata in another compression operation, even with an entirely different compression system, and included as information used during the first pass compression operation for that another compression operation. *Id.* This will enhance the speed of the another compression operation due to the existence of the metadata that would not otherwise be available, since the patterns in the metadata that exist after those passes are indicative of more subtle redundancies or partial redundancies that are not otherwise easily apparent.

At the end of a successful compression routine for a stream of data, a series of compressed blocks will result. *Id. at page 21, lines 4-21.* The decompression routine, however can perform the inverse decompression operation in a single pass. As a result, while compression may require several compression passes on the same data, decompression can always be achieved in a single pass, since the original data stream can be derived from finally compressed data. *Id. at page 21, lines 9-21.*

VI. ISSUES ON APPEAL

(37 C.F.R. 1.192(c)(6))

Whether claims 1-3, 6-7, 16-18, 20-21, 29, 39-53, 61, 64 and 66-73 are anticipated under 35 USC §102(e) in view of US Patent No. 6,366,289 to Johns (herein "Johns")?

Weather claims 4-5, 11-13, 19, 54-60, 62-63 and 65 are obvious, under 35 USC § 103(a), based on Johns taken in view of US Patent No. 5,586,280 to Simms (herein "Simms")?

Weather claims 8-10 and 14-15 are obvious, under 35 USC § 103(a), based on Johns taken in view of US Patent No. 6,043,897 to Morikawa et al. (herein "Morikawa")?

VII. GROUPING OF CLAIMS

(37 C.F.R. 1.192(c)(7))

The rejected claims have been grouped as follows:

1. Claims 1, 2, 6, 7, 11, 12, 13, 16, 21, 47-53, 55-59, and 69-73 stand or fall together and should not be grouped with the other claims;
2. Claims 3-5, 19, 54, 60, 63, and 65 stand or fall together and should not be grouped with the other claims;
3. Claims 8-10, 14, and 15 stand or fall together and should not be grouped with the other claims;
4. Claims 17, 18, and 64 stand or fall together and should not be grouped with the other claims;
5. Claims 20 and 66 stand or fall together and should not be grouped with the other claims;
6. Claims 29, 67, and 68 stand or fall together and should not be grouped with the other claims;
7. Claims 39 and 61 stand or fall together and should not be grouped with the other claims;
8. Claims 40-43 stand or fall together and should not be grouped with the other claims; and
9. Claim 46 stands alone and should not be grouped with the other claims.

The grouping of the claims will become clear in view of the arguments presented below.

VIII. ARGUMENTS

(37 C.F.R. 1.192(c)(8))

The Examiner rejected claims 1-3, 6-7, 16-18, 20-21, 29, 39-53, 61, 64 and 66-73 under 35 USC §102(e) as being anticipated over Johns. Additionally, the Examiner rejected claims 4-5, 11-13, 19, 54-60, 62-63 and 65 under 35 USC § 103(a) as being unpatentable over Johns in view of Simms. Finally, the Examiner rejected claims 8-10 and 14-15 under 35 USC § 103(a) as being unpatentable over Johns in view of Morikawa. Among the rejected claims are independent claims 1, 39, 40, and 46.

A. Johns does not teach, disclose, or suggest each and every element of independent claims 1, 39, 40 and 46.

The Examiner rejected independent claims 1, 39, 40, and 46 under 35 USC § 102(e) as allegedly being anticipated by Johns. Appellant asserts that Examiner's rejection based on Johns is flawed for at least two reasons: *First,* the Examiner has misunderstood Appellant's invention as claimed; *Second,* in light of the misinterpretation the Examiner has misapplied the Johns reference. Accordingly, it is Appellants position that the rejection based on Johns is wrong.

1. The Examiner has misinterpreted Appellant's independent claim 1 and in light of that misinterpretation has improperly concluded that Johns anticipates independent claim 1.

Claim 1 recites

A method of operating upon digital data comprising the steps of:

partitioning the digital data into a plurality of blocks;

creating a plurality of first threads, such that each first thread includes at least one of the plurality of blocks; and

operating upon each of the plurality of first threads to obtain a plurality of compressed first threads, each compressed first thread including at least one compressed block of digital data.

The Examiner has misinterpreted the term "thread" as used in this claim. The reasoning for doing so is evident: using an inappropriate interpretation makes possible an inappropriate characterization of the prior art.

The word thread, as used in computer science, is widely understood as a sequence of program instructions. The specification usage is consistent with this ordinary usage of the word "thread," and makes it clear that this is the only usage intended. For example, the specification teaches that the "C/D [compression/decompression] engine 230 being implemented using at least one processor that has the capability of operating multiple threads simultaneously." Still further, a number of different processors in parallel can be used to even more efficiently implement the C/D engine 230. No matter which implementations are used, the controller interface and the C/D engine 230 are preferably implemented as a sequence of program instructions, written in C++ or some other computer language or, alternatively, implemented in hardware. (page 7, lines 4-10).

Consistent with the ordinary usage of the word "thread" in this computer science context, processor operations operate upon data, and that is explicitly referred to in claim 1: "each first thread includes at least one of the plurality of blocks," with the blocks being digital data. Thus, it is clear that threads are explicitly recited, as are the blocks of data associated with each thread.

In contrast to this construction, the Examiner reads "each first thread includes at least one of the plurality of blocks" as meaning that a thread is the same as or "reads on" chunks. *See Office Action dated September 25, 2003 at paragraph 3, page 3.* Hence, given an inappropriate construction "threads" the Examiner reaches the conclusion that "threads" are the same as chunks of data and, hence, Johns anticipates claim 1.

Not only is this reasoning improper, but it is not supported by Johns. Johns specifically teaches -- consistent with ordinary usage in the art -- that a thread is related to related to a "process" (i.e. sequence of instructions) at column 15, lines 35-39. Further, John's teaches that this process (or thread) operates upon "chunks." As such, Johns refers to a thread in the conventional manner known by one of ordinary skill in the art. Given Johns teaching, and differentiation between the use of the word thread and chunks, at best Johns teaches using two threads (a background thread and a foreground thread), with the background thread having chunks of data associated with it. Johns

does not teach a plurality of threads with each thread including a plurality of blocks, and then operating on each of the plurality of threads.

2. The Examiner has misinterpreted Appellant's independent claims 39 and 40, which recite multiple passes for compression and a single pass for decompression and in light of that misinterpretation has improperly concluded that Johns anticipates independent claims 39 and 40.

With respect to claims 39 and 40, Johns does not teach nor suggest compressing "using multiple passes" and decompression "using a single pass" as set forth in independent claims 39, 40, and 46.

The term "pass" in the application means the number of times that a specific portion of data is operated upon in a compression or decompression process (such that if two compression operation are performed on the specific portion of data , this is two passes, and if that same portion of data has a third compression operation performed on it, then that is three passes), as is apparent from, for example, the "passes required" variable in a compression process indicating the "the number of passes that the C/D controller estimates will be required" and in Figure 3B, which shows the step of "continue with routine at 420 for a subsequent pass." Further, in the decompression process the description of decompression "in a single pass," meaning that no matter how many times a portion of data was compressed, it will be decompressed in a single pass. "Pass" in both the context of compression and decompression is therefore used in the same manner.

The Examiner, however, is not using the term pass in the same manner for both compression and decompression. Rather the Examiner has quoted from Johns as follows:

> [t]he invention provides a method for managing display memory where a visible display image as well as other images are partially compressed and partially uncompressed. Portions of each image that are accessed frequently are maintained...

Col. 5, lines 45-48. The Examiner goes on to conclude that the above referenced portion of the Johns teaches or suggests "multiple passes" for compressing and a "single pass" for decompressing. This conclusion, however, is based upon an assertion that it is "inherent" that compression may be

handled in multiple passes, since "each newly compressed section is 'added' to the compressed file." This interpretation of "multiple passes" as it relates to compression is inappropriate, since, as explained above. That this interpretation is inappropriate is further illustrated by the interpretation of "pass" differently as it relates to decompression.

With the inconsistent usage of "pass," the Examiner is of course able to assert that John's anticipates these claims, but applicant respectfully asserts that no such anticipation exists, as Johns clearly does not teach or suggest the invention set forth in these claims.

The manner in which the present rejection has evolved is also significant, as it illustrates the thin thread that the Examiner is relying on in order to maintain the anticipation rejection. In the Official Action dated January 2, 2003, at page 7, rejected claim 40 based on the following reference to Johns:

> The invention provides a method for managing display memory where a visible display image as well as other images are partially compressed and partially uncompressed.

In response, Appellant identified the problem with this rejection in the Remarks section of Appellant's response to the January Official Action, wherein Appellant states that "[n]othing is mentioned about compressing in a multiple passes to obtain compressed digital data [and n]othing is mentioned about decompressing the compressed digital data in a single pass." *See Appellant's Response dated July 2, 2003, at page 21.*

In the following and final Official Action, dated September 25, 2003, the Examiner again maintained the same rejection and in the Remarks section of Examiner's Official Action, at page 21, the Examiner concludes that:

> Compression may be handled in multiple passes, wherein each newly compressed section is added to the compressed file, and it is also inherent that a single compressed file (which may have been obtained in multiple compression passes) can be decompressed in a single pass . . . .

It appears that once the meritless rejection was identified and pointed in Appellant's Remarks dated July 2, 2003, at page 21, the Examiner simply responded in the September 25, 2003 Final Official Action by asserting that compression in multiple passes and decompression in a single pass is inherent, without a basis for making such a statement.

3. In an initial Official Action, the art cited by the Examiner failed to anticipated Appellant's claim 40 and, once the Examiner's misunderstanding and the error with the rejection was identified in the Final Office Action, the Examiner's then asserted that the teaching of claim 40 are inherent: in light of this newly found inherency conclusion, he then improperly concluded that Johns anticipates independent claim 40.

Appellant's claim 40 is directed to means for compressing the digital data and means for decompressing the compressed digital data. The structure disclosed in Appellant's specification for compression and decompression includes a detailed description of a compression/decompression (C/D) engine 230. *See Appellant's Specification beginning at page 6, line 10.* While the C/D engine 230 is described as being implementable in many ways, the description clearly uses "threads" in the context as has been described above, teaching for example, that the C/D engine 230 can include "at least one processor capable operation upon multiple threads simultaneously" or "a number of different processors in parallel can be used to even more efficiently implement the C/D engine 230." Appellant's specification sets forth that the decompression algorithm is an inverse of the compression algorithm with one significant difference: whereas the compression process takes multiple passes, the decompression process can always occur in a single pass. *See Appellant's Specification at page 21. lines 9-21.* As a result, in addition to the arguments made in Section 2, *supra*, it is even more apparent that claim 40 should be interpreted in the manner described, since claim 40 expressly recites these elements in means plus function form, which requires reference to the specification to properly interpret the structure and function of the recited means elements. *See Gechter v. Davidson* 116 F3d 1454, 43 USPQ2d 1030 (Fed. Cir. 1997) stating that the scope of the claimed means must be construed in light of the disclosure in the specification as well as the exact function of the means in order to determine if the cited art discloses the same means. Accordingly, for this reason, in addition to the reasons set forth above with respect to claim 39, claim 40 should be allowed.

4. The Examiner has misinterpreted Appellant's independent claim 46 and in light of that misinterpretation has improperly concluded that Johns anticipates independent claim 46.

With respect to independent claim 46, the Examiner's reference to column 6, lines 1-6 for the teaching of metadata that includes a "representation of patterns" is misplaced, as there is no teaching or suggestion of such a representation of patterns in this or any other section of Johns.[1] The Examiner's rejection of claim 46 is a mere conclusion that is unsupported by Johns. Appellants' claim 46 specifically relates to making a compression system more efficient by

> "obtaining metadata representative of patterns in first digital data obtained from the compression of the first digital data in a first compression system and distributing the metadata to the at least a second compression system so that the second compression system can use the metadata to compress second digital data which the second compression system needs to compress." *See Appellant's claim 46.*

The metadata referred to in claim 46 is information that represents a pattern in digital data in one compression system that is used in a second compression system for compressing a second digital data. Generally stated, the metadata relating to patterns resulting from compression in the first system is used in compression of data in a second system.

The Examiner asserts that Johns teaches the foregoing concept of using metadata related to compression patterns in one system for compression in a second system. However, Appellant has not found such a reference in Johns, even when the cited portions are considered with the broadest possible interpretation. Johns, as cited by the Examiner, teaches that the "chunk may be compressed or uncompressed using a lossless or lossy compression method." Johns goes on to teach that "the relationship between a conceptual two-dimensional image (the virtual frame buffer) and a physical address space, where the discrete chunks of the display image are distributed randomly." The foregoing teaching of Johns does not teach obtaining metadata that represents a pattern in one system for use in another system. Johns only teaches that there is information about

1 It is also worth noting that the Examiner has indicated that claim 26, which is also related to metadata, contains allowable subject matter because the prior art of record "do not disclose, teach, or suggest the claimed invention of (in combination with all other features in the claim) . . . as claimed in claim 26." *See Official Action dated September 23, 2003.*

the way chunks can be compressed and decompressed and that there is a relationship between two-dimensional image and a physical address space where chunks are randomly distributed. This teaching does not anticipate Appellant's invention as set forth in claim 46. As such, the rejection of claim 46 is improper.

B. The Examiner has improperly rejected the dependent claims under 35 USC 102.

1. The cited art does not anticipate claim 3.

Johns does not teach or disclose the concept of threads that include data in light of independently operating upon each of the plurality of first threads. The portions of Johns cited by the Examiner are directed toward two entirely different concepts, those being (1) that "the compression process can treat each coefficient independently" and (2) "the compression methods described above compress images in independent blocks" within the context of data compression. While each uses the word "independently," as does claim 3, this word usage does not teach or suggest independently operating upon a plurality of threads to obtain compressed data associated with each different thread. Thus, claim 3 was improperly rejected.

2. The cited art does not anticipate claim 6.

The Examiner's misinterpretation, as discussed above with respect to claim 1, is again demonstrated in the rejection of claim 6. More specifically, the Examiner has indicated that "'combining the plurality of compressed first threads' is read on 'compressed chunks are linked together in a linked list format'." *See Final Official Action, page 3.* In addition to the interpretational differences argued by Applicant above with respect to "thread," this rejection further illustrates the Examiner's error relating to the threads. In particular, the claim requires "combining" compressed blocks from the threads after the compression operation using the plurality of threads, as recited. In contrast, the Examiner points to an operation in Johns that creates a linked list of various chunks of data, but these various chunks of data have not been previously

separated and associated with various threads, and then operated upon as recited. Thus, claim 6 contains allowable subject matter.

3. The cited art does not anticipate claim 7.

The Examiner rejected claim 7 based on the same misinterpretation and misapplication of Johns. If properly interpreted, claim 7 is patentably distinct.

Even if the Examiner's interpretation is used, however, claim 7 is allowable. Appellant's claim 7 recites that "the step of creating the plurality of first threads includes the step of <u>*associating each of the plurality of blocks of digital data with one of the plurality of first threads.*</u>" This relates to a distinct association between each thread and a block of data. In contrast, the teaching of Johns, which the Examiner relies on to reject claim 7, specifically states that the VFB controller "computes the address of the compressed block control data associated with the chunk and reads this control data from the memory device where it resides." *See Johns, col. 7 at lines 62-66.* It is clear that Johns teaches an association between the compressed block control data and the chunk; in other words data is being associated with data. Further, however, this association is being performed <u>after</u> compression has already taken place. In contrast, as recited, in the step of creating the first plurality of threads, data is being associated with different threads <u>prior to</u> it being compressed, and as such Johns does not anticipate this claim.

4. The cited art does not anticipate claim 17

Claim 17 recites additional limitations not present in claim 1, which relate to further compressing blocks that have already been previously compressed using the processes described in claim 1. These additional steps are recited as:

> operating upon each of the compressed first threads to eliminate each of the compressed first threads and retain the compressed first blocks;
> creating a plurality of second threads, such that each second thread includes at least one of the plurality of compressed first blocks; and
> operating upon each of the plurality of second threads to obtain a plurality of compressed second threads, each compressed second thread including at least one compressed second block of digital data.

Appellants' claim 17 is directed to a method of further compressing already compressed blocks. The Examiner has relied upon the misinterpretation of the claim language to conclude that Johns teaches each and every element of Appellant's claim 17 even though Johns clearly does not teach further compressing already compressed blocks. The rejection is accomplished by citing various aspects of Johns out of context.

For example, the Examiner assumes that word "eliminate" reads on "freeing up memory" and based on that assumption concludes that Johns teaches "operating upon each of the compressed first threads to eliminate each of the compressed first threads and retain the compressed first blocks." The Examiner's analysis is a mere conclusion without reason, since the portion of Johns that the Examiner cites specifically teaches:

> "[t]o improve performance, the controller leaves blocks in an uncompressed format until it is necessary to free up memory. The system compresses least recently used blocks when necessary to free up memory for other decompressed blocks."

Thus, Johns teaches that in order to free up memory the least recently used blocks are compressed and nothing more.

The Examiner also concludes that plurality of (second) threads reads on subregions called chunks. The cited portion of Johns that Examiner relies upon that relates to the subregions called chunks specifically teaches "managing compressed and uncompressed pixel memory . . . that manages images in two-dimensional image subregions called chunks." Generally stated, Johns teaches that managing subregion called a chunk that contains compressed and uncompressed pixel memory. In contrast, Appellant's claim 17 recites "creating a plurality of second threads that include at least one of the plurality of compressed blocks . . ." Thus, managing a chunk that has compressed and uncompressed pixel memory, as taught by Johns, is not the same as creating second threads that each include, among other things, previously compressed blocks, as set forth in Appellant's claim 17. Consequently, the Examiner's conclusion that this element of Appellant's claim is anticipated by Johns is a mere conclusion and incorrect.

The Examiner also concludes that obtaining compressed second threads that include at least one compressed second block is taught by Johns. The cited portions of Johns allegedly teaching the concept of compressed and uncompressed chunks are col. 7, lines 62-66 and col. 16, lines 8-13; this

is not the same as operating on the second threads to obtain twice compressed blocks, recited as the "at least one compressed second block of digital data." Accordingly, Johns does not teach this element of Appellant's claim 17.

It is worth noting that there is no consistency between the various teaching that the examiner relies upon for teaching these different recited steps. Thus, viewed together, it is apparent that Johns does not teach or suggest the invention of claim 17.

5. The cited art does not anticipate claim 20.

Claim 20 additionally recites that the compression algorithm used to operate upon "each block" as recited in claim 1 is also used to operate upon the "corresponding compressed block (the compressed first blocks.) Thus, the same compression algorithm is used to further compress an already compressed block of data.

Generally, conventional compression techniques do not attempt to further compress already compressed data. Conventional techniques will compress uncompressed source data different times, each time using a different compression algorithm, to determine which compresses the best, but that is not what is recited here. Thus, the Examiner's reference to Johns using "a variety of compression and decompression methods." *See Johns col. 17, lines 45-51,* is not sufficient to teach the patentable subject matter of claim 20. Johns does not anywhere teach or suggest further compressing already compressed data, and as such cannot teach or suggest the particular manner recited by applicant to do this: using a plurality of first threads to obtain compressed blocks (from claim 1), then using a plurality of second threads to obtain second compressed blocks (from claim 17, and, in claim 20, ensuring that the compression technique used in obtaining the compressed blocks is also used in obtaining the second compressed blocks. Despite the Examiner's reference to a large number of pages in Johns (*See Johns col. 17, line 53 – col. 20, line 45*) there is no such teaching there. Thus, the Examiner's conclusion is clearly without support and the Examiner's rejection of claim 20 is improper.

6. The cited art does not anticipate claim 44.

Appellant's claim 44, ultimately dependent on claim 40 which recites "means for compressing the digital data using multiple passes" as described above, is further dependent on claim 41 where the means for compressing is described as including a interface controller and a compression engine. Claim 43 further describes this compression engine as comprising a plurality of processing units, and claim 44, derivatively depend on claim 43, recites that "each of the plurality of central processing units operate upon a different plurality of threads." *See Appellant's claim 44 as originally filed.*

The Examiner has rejected claim 44 based on a mere finding that the teaching of Johns mentions "implementation in other computer system configurations, including . . . multiple processor systems." *See Johns col. 4, lines 20-23.* Thus, Johns only teaches that multiple processor systems can be used in the context of a computing environment that can implement various concepts related to display architecture and control of a computer. Johns does not even mention the concept of multiple processors operating upon a different plurality of threads as recited. In order to reach Examiner's improper conclusion that claim 44 is anticipated by Johns, the Examiner asserts that "it is inherent that 'other computer system configurations' operate upon different threads." *See Final Official Action dated September 25, 2003.* While it may be inherent to use multiple processors in certain computer systems, it is not inherent to operate upon a different plurality of threads which, as recited, are all used for compression, and none of the art teaches or suggests such a feature. Accordingly, Appellant's claim 44 is not inherent to one of ordinary skill in the art.

7. The cited art does not anticipate claims 47, 55, and 69.

Appellant's claims 47, 55, and 69 set forth that the thread includes control signals.[2] The Examiner asserts that "control signal" reads on "control data" as disclosed in Johns. This is an improper reading of Appellant's claim language and Johns. Johns teaches that in order to access compressed chunks or data the "controller computes the address of the compressed block control data associated with the chunk . . ." Thus, Johns specifically teaches that the compression block

2 The Examiner has rejected claim 55 under 35 USC 103, while 47 and 69 were rejected under 35 USC 102. However, the Examiner has not provided any reason for the 103 rejection, which restates the exact same language.

control data is associated with the compressed chunks and that compression control data specifically contains information relating to the location of the data and if the data at that location is compressed or uncompressed. *See Johns at col. 7, lines 12-15.* In contrast, Appellant's claims 47, 55, 69 set forth that the thread includes control signals, wherein the control signals relate specifically to the control of the compression process. Thus, the term "compressed clock control data" as used in Johns is directed at information about the location and compression state of the data whereas the "control signals" as used in Appellant's claims 47, 55, and 69 are related to controlling the manner the thread uses to compress and, hence, are not anticipated or made obvious by Johns. Accordingly, the Examiner's rejection of claims 44 and 69 as being anticipated by Johns and claim 55 as being obvious in light of Johns were improper.

8. The cited art does not anticipate claims 51- 53.

Whereas claim 7 recites that each of the plurality of first threads share certain common compression characteristics, which is patentable as argued previously, in addition, claim 53 recites that different first threads include blocks of data that do not share common compression characteristics. This is stated in different terms with respect to claims 51 and 52, which recite that different first threads include blocks of data containing different types of data. This advantageously allows for different threads to refer to different program instructions for different compression routines, and is not disclosed or suggested in any of the cited references. Accordingly, claims 51-53 were improperly rejected.

9. The cited art does not anticipate claim 61.

Claim 61 explicitly combines the limitations from claim 1, referenced above, with the limitations of multiple pass compression and single pass decompression from the claim on which it is dependent, independent claim 39. As such, this combination is patentably distinct, as there is not a single reference that teaches or suggests multiple pass compression and single pass decompression, and the usage of a plurality of threads for the step of compressing. As such, this combination is patentably distinct.

10. The cited art does not anticipate claim 64 and, hence, claim 64.

Claim 64 explicitly combines the limitations from claim 17, referenced above, with the limitations from claim 61, also referenced above. This combination is patentably distinct, as there is not a single reference that teaches or suggests multiple pass compression and single pass decompression, and the usage of a plurality of first and second threads for the steps of compressing in two different passes. As such, this combination is patentably distinct and, hence, claim 64 was improperly rejected.

C. The Examiner has improperly rejected the dependent claims under 35 USC 103.

1. The combination of references suggested by the Examiner do not suggest, disclose, or make obvious claim 4.

The Examiner's rejection of claim 4 further demonstrates Examiner's misinterpretation of Appellant's invention as set out in claim 4. More specifically, the Examiner has indicated that Johns does not teach "threads are independently operated upon in parallel." Additionally, Simms does not teach operating upon threads in parallel. Accordingly, the Examiner assertion that the motivation to combine in order to establish obviousness is based upon "knowledge generally available in the art" because that "would increase efficiency and reduce operation time" is wholly speculative and unsupported. The mere fact that such parallel operations could potentially increase efficiency and reduce operation time is not sufficient because such as a conclusion does not explain the increase complexity and solution for potential problems that may cause malfunction. Thus, on the established record there is no support for the Examiner's assertion that such a motivation would have been part of generally available knowledge as this only takes into account the end result and not how that end result is achieved.

Furthermore, the Examiner has again relied upon the misinterpretation that threads are the same as data to conclude that Simms teaches "certain ones of first threads are independently operated upon in parallel." However, the portion of Simms that the Examiner relies upon is unrelated to threads; the portion of Simms relates specifically to data. It is clear, as the Examiner has referenced, that Simms teaches that "[f]or data which is being compressed, in parallel with

organization of compressed data into groups . . ." *See Simms, col. 17, lines 16-20.* Thus, based on the foregoing, Simms teaches that compression of data can occur while the compressed data is organized into groups. However, this is unrelated to the concept of independently operation upon certain first threads in parallel; compression of data in parallel with organization of compressed data into groups is not the same as independently operation upon certain threads in parallel. It is also clear, again as the Examiner has referenced, that Simms teaches in "parallel with the generation of the main data blocks, 35-byte sub-data blocks are also generated . . ." *See Simms, col. 19, lines 27-28.* However, it is unclear how generation of data in parallel is related to independently operating upon certain threads in parallel. The only way that the Examiner can reach that conclusion that modifying "Johns by the teaching of Simms" would have been obvious is to first conclude that a thread is the same data. This is a clear misinterpretation of the Appellants' claim 4, which states that threads include data and threads are independently operated upon in parallel.

2. The combination of references as suggested by the Examiner do not suggest, disclose, or make obvious claims 8, 10 and 14.

The Examiner's assertion that Morikawa teaches "the step of predicting and estimated compression time (see column 2, lines 14-18) and estimated compression amount for each block (see column 5, lines 57-63)" is based upon a misinterpretation of Morikawa.

Initially, Morikawa is combined with Johns in a manner where there would not have been motivation to do so. Morikawa is a specific technique used in a device used to print on paper an electronic document, which is not the concern of Johns at all. As such, one of ordinary skill in the art would not have attempted the combination suggested by the Examiner, and this illustrates that inappropriate hindsight is being used.

Assuming *arguendo* that the art did teach the modification of Johns by Morikawa, such a combination still does not teach, disclose, or suggest Appellant's invention as set forth in claims 8, 10 and 14. In particular, Morikawa, however, measures the compression time for one block in order to estimate the time for compressing all of the blocks, which is not the same as "estimating compression time . . . for each block" as set forth in Appellant's claims 8, 10 and 14.

Similarly, Morikawa teaches actually measuring and storing information related to "size of compressed image and measured compression time" without any reference to estimating compression amount. *See id, col. 5, lines 45-63.* Thus, measuring and storing information related to actual "size of compressed image", as taught by Morikawa, is not the same as "estimating compression amount for each block" as set forth in Appellant's claims 8, 10 and 14.

Thus, the Examiner's rejection of claims 8, 10 and 14 is improper.

3. The combination of references as suggested by the Examiner do not suggest, disclose, or make obvious claims 9 and 15.

The Examiner's rejection of Appellant's claims 9 and 15 is improper because the art can not be combined as suggested by the Examiner and, if assuming arguendo that such a combination were allowed then, the combination does not teach, suggest, or disclose the invention as set forth in claims 9 and 15. More specifically, Examiner's rejection further demonstrates that the Examiner has not only misinterpreted Appellant's claim 9, but also that the Examiner has failed to identify where all limitations of the claimed invention are met by the cited prior art references. *See requirement MPEP § 2142 at 2100-121 (Aug. 2001); In re Vaeck, 947 F.2d 488, 20 U.S.P.Q.2d 1438 (Fed. Cir. 1991).* As indicated, Morikawa teaches the concept of obtaining actual values for each block and predicting an estimate time for repeating the same action for all of the blocks as well as actually measuring and storing information related to "size of compressed image and measured compression time" without any reference to estimating compression amount.. *See Morikawa col. 2, lines 14-18 and col. 5 lines 45-63.* There is no teaching in the art relating to determining "which blocks should be associated with the same first thread" based on the estimate compression time and amount. It is only because of Examiner's misinterpretation of Appellant's claim language that the Examiner is able to conclude, based on the improper reading of Morikawa (discussed supra), that the art teaches "which blocks should be associated with the same first thread." Thus, Examiner's rejection of Appellant's claim 9 and 15 as being obvious is a further demonstrates the Examiner's misinterpretation of Appellant's claim language and, thus, and improper rejection of Appellant's claims 9 and 15.

4. The combination of references as suggested by the Examiner do not suggest, disclose, or make obvious claim 11.

Appellant's claim 11 is not obvious in light of Johns as modified by Simms. Claim 11 teaches the

> "step of creating each of the plurality of first threads uses a data type of each of the plurality of blocks so that each of the first threads contains blocks which have a similar data type."

Additionally, neither Johns nor Simms contain "threads", for the reasons recited, the Examiner further asserts that Simms teaches

> "the step of creating each of the plurality of first threads uses a data type of each of the plurality of blocks so that each of the first threads contains blocks which have a similar data type (see column 7, lines 11-16)."

In short the Examiner has merely recited Appellant's claim language and provide a citation to Simms and concluded that this would have been obvious because it would "enable the system for categorize data into blocks of data with common characteristics among the data items." *See Official Action dated September 25, 2003, page 11, third paragraph.* The cited portion of Simms, however, does not support Examiner's rejection given that the cited portion specifically states that

> "entries in the block access table each comprise a FLAG entry indicating the type of the entry and a COUNT entry indicating its value. The FLAG field is 8 bits and the COUNT filed is 24 bits. The bits in the FLAG field have the following significance . . ."

Accordingly, the subject matter of claim 11 is patentably distinct.

5. The combination of references as suggested by the Examiner do not suggest, disclose, or make obvious claim 16.

Claim 16 recites that in determining the size of each of the plurality of blocks, the data type of each block is taken into account. This allows for different blocks of different sizes, depending upon the type of data that they contain. The cited art does not teach or suggest partitioning blocks of different sizes based upon the data type. As such, this feature is patentably distinct.

IX. Conclusion

For the reasons advanced above, Appellant concludes the following:

the rejection of claims 1-3, 6-7, 16-18, 20-21, 29, 39-53, 61, 64 and 66-73 as being anticipated under 35 USC §102(e) in view of Johns is improper;

the rejection of claims 4-5, 11-13, 19, 54-60, 62-63 and 65 for being obvious under 35 USC §103(a) based on Johns taken in view of Simms was improper; and

the rejection of claims 8-10 and 14-15 for being obvious under 35 USC § 103(a) based on Johns in view of Morikawa was improper.

Accordingly, Appellant respectfully submits that the Examiner's Final Rejection on September 25, 2003 should be reversed and all claims allowed by this Honorable Board.

Respectfully submitted,
PILLSBURY WINTHROP LLP

By [signature]
David A. Jakopin, Reg. No. 32,995
Tel: 650-233-4790

Reply to Customer No. 27498
2475 Hanover Street
Palo Alto, CA 94304

60377355

IX. APPENDIX OF CLAIMS ON APPEAL

(37 C.F.R. 1.192(c)(8))

The following claims are on appeal:

1. **(Original)** A method of operating upon digital data comprising the steps of:

partitioning the digital data into a plurality of blocks;

creating a plurality of first threads, such that each first thread includes at least one of the plurality of blocks; and

operating upon each of the plurality of first threads to obtain a plurality of compressed first threads, each compressed first thread including at least one compressed block of digital data.

2. **(Original)** A method according to claim 1 wherein the step of operating upon each of the first threads performs lossless compression.

3. **(Original)** A method according to claim 1 wherein the step of operating upon each of the first threads independently operates upon each of the plurality of first threads.

4. **(Original)** A method according to claim 1 wherein at least certain ones of the first threads are independently operated upon in parallel.

5. **(Original)** A method according to claim 4 wherein, during the step of operating, at least two different compression algorithms are used to independently operate upon different first threads.

6. **(Previously Amended)** A method according to claim 1 further comprising the step of combining compressed blocks in each of the plurality of compressed first threads to obtain digitally compressed data.

7. **(Original)** A method according to claim 1 wherein the step of creating the plurality of first threads includes the step of associating each of the plurality of blocks of digital data with one of the

plurality of first threads such that blocks within each of the plurality of first threads share certain common compression characteristics.

8. **(Original)** A method according to claim 7 further including the step of predicting an estimated compression time and estimated compression amount for each block.

9. **(Original)** A method according to claim 8 wherein the step of creating the plurality of first threads also uses estimated compression time and estimated compression amount to determine which blocks should be associated with the same first thread.

10. **(Original)** A method according to claim 8 wherein the estimated compression time and estimated compression amount are made based upon a selected compression algorithm, and wherein the step of predicting includes the step of determining whether a proposed estimated completion time that is based upon one of the compression algorithms available for selection will allow for a desired compression amount to be achieved within a desired compression time for the digital data.

11. **(Original)** A method according to claim 1 wherein the step of creating each of the plurality of first threads uses a data type of each of the plurality of blocks so that each of the first threads contains blocks which have a similar data type.

12. **(Original)** A method according to claim 11 wherein the data type is determined according to header information related to each block.

13. **(Original)** A method according to claim 11 where the data type is determined by comparing the block data to various predetermined data patterns.

14. **(Original)** A method according to claim 1 further including the step of predicting an estimated compression time and estimated compression amount for each block.

15. **(Original)** A method according to claim 14 wherein the step of creating the plurality of first threads uses estimated compression time and estimated compression amount to determine which blocks should be associated with the same first thread.

16. **(Original)** A method according to claim 1 wherein the step of partitioning data includes the step of determining a size of each of the plurality of blocks taking data type of each block into account.

17. **(Original)** A method according to claim 1 further including the steps of:

operating upon each of the compressed first threads to eliminate each of the compressed first threads and retain the compressed first blocks;

creating a plurality of second threads, such that each second thread includes at least one of the plurality of compressed first blocks; and

operating upon each of the plurality of second threads to obtain a plurality of compressed second threads, each compressed second thread including at least one compressed second block of digital data.

18. **(Original)** A method according to claim 17 wherein the step of operating upon each of the second threads independently operates upon each of the plurality of second threads.

19. **(Original)** A method according to claim 17 wherein at least certain ones of the second threads are independently operated upon in parallel.

20. **(Original)** A method according to claim 17 wherein, during the step of operating upon each of the plurality of second threads, the same compression algorithm used to operate upon each block is also used to operate upon the corresponding compressed block.

21. **(Previously Amended)** A method according to claim 17 further comprising the step of combining the compressed blocks in each of the plurality of compressed second threads to obtain digitally compressed data.

22. **(Original)** A method according to claim 17 wherein the step of creating the plurality of second threads includes the step of associating each of the plurality of compressed first blocks with one of the plurality of second threads such that compressed first blocks within each of the plurality of second threads share certain common compression characteristics.

23. **(Original)** A method according to claim 22 wherein each of the second threads contains compressed first blocks that were created from the same first thread.

24. **(Original)** A method according to claim 23 wherein the number of second threads is greater than the number of first threads.

25. **(Original)** A method according to claim 22 wherein compressed first blocks that were within the one of the first threads are used to form two distinct second threads.

26. **(Original)** A method according to claim 17 wherein the step of operating upon each of the plurality of first threads also results in obtaining a plurality of first metadata sets, each first metadata set including portions of compressed first blocks which are determined to possibly have redundancies disposed therein.

27. **(Original)** A method according to claim 26 wherein the step of operating upon each of the first threads will maintain for each thread a pattern of data in an initial compressed first block that corresponds to a first metadata pattern, each of the different first metadata patterns for each first thread combining to result in the first metadata set for that first thread.

28. **(Original)** A method according to claim 27 wherein the pattern of data in the initial compressed first block is maintained during subsequent steps.

29. **(Original)** A method according to claim 17 wherein each first thread has an associated first metadata set.

30. **(Original)** A method according to claim 26 wherein the step of creating the plurality of second threads includes the steps of:

determining which compressed first blocks should be associated with the same second thread; and

using the first metadata sets to eliminate redundancies in some of the compressed first blocks associated with at least some of the second threads.

31. **(Original)** A method according to claim 30 wherein the step of operating upon each of the plurality of second threads also results in obtaining a plurality of second metadata sets, each second metadata set including portions of compressed second blocks which are determined to possibly have redundancies disposed therein.

32. **(Original)** A method according to claim 31 wherein the second metadata set is a subset of the first metadata set.

33. **(Original)** A method according to claim 26 wherein:

the step of operating upon each of the first threads will maintain for each thread a pattern of data in an initial compressed first block that corresponds to a first metadata pattern, each of the different first metadata patterns for each first thread combining to result in the first metadata set for that first thread;

the step of using the first metadata sets to eliminate redundancies in some of the compressed first blocks associated with at least some of the second threads maintains the pattern of data in the initial compressed first block and eliminates the pattern of data in a subsequently compressed first block.

34. **(Original)** A method according to claim 33 wherein, during the step of using the first metadata sets to eliminate redundancies in some of the compressed first blocks associated with at least some of the second threads, the pattern of data in the subsequently compressed first block is replaced with a pointer and an operation designator, thereby obtaining a plurality of compressed and reduced first blocks in each second thread.

35. **(Original)** A method according to claim 34 wherein, during the step of operating upon each of the plurality of second threads, the same compression algorithm used to operate upon each block is also used to operate upon the corresponding compressed and reduced first block to thereby obtain the compressed second blocks.

36. **(Original)** A method according to claim 35 wherein the operation designator identifies an operation used to eliminate redundancies in the first compressed data blocks, the operation being one of an equal to comparison operation, a greater than or equal to comparison, and a less than or equal to comparison.

37. **(Original)** A method according to claim 36 wherein, during the step of operating upon each of the plurality of second threads, the comparison operation selected is adaptively determined.

38. **(Original)** A method according to claim 37 wherein the adaptive determination is made based upon the pattern of the compressed blocks as compared to representative file type patterns.

39. **(Original)** A method of operating upon digital data comprising the steps of:
compressing the digital data using multiple passes of a predetermined compression algorithm to obtain compressed digital data; and
decompressing the compressed digital data using a single pass of a corresponding decompression algorithm to obtain the digital data.

40. **(Original)** An apparatus for operating upon digital data comprising the steps of:

means for compressing the digital data using multiple passes of a predetermined compression algorithm to obtain compressed digital data; and
means for decompressing the compressed digital data using a single pass of a corresponding decompression algorithm to obtain the digital data.

(Original) An apparatus according to claim 40 wherein the means for compressing includes:

an interface controller; and

a compression engine.

(Previously Amended) An apparatus according to claim 41 wherein the compression engine comprises a single central processing unit.

(Previously Amended) An apparatus according to claim 41 wherein the compression engine comprises a plurality of central processing units.

(Previously Amended) An apparatus according to claim 43 wherein each of the plurality of central processing units operate upon a different plurality of threads.

(Original) An apparatus according to claim 44 wherein the plurality of central processing units comprise a plurality of digital signal processors.

(Original) A method of allowing a plurality of compression systems to operate more efficiently comprising the steps of:
obtaining metadata representative of patterns in first digital data obtained from the compression of the first digital data in a first compression system; and
distributing the metadata to the at least a second compression system so that the second compression system can use the metadata to compress second digital data which the second compression system needs to compress.

47. **(Previously Presented)** A method according to claim 1 wherein each first thread further includes control signals.

48. **(Previously Presented)** A method according to claim 47 wherein the control signals in each first thread include a compression routine control signal indicating a compression routine to be used in the step of operating.

49. **(Previously Presented)** A method according to claim 48 wherein different ones of the compression routine control signals indicate different compressions routines for different first threads.

50. **(Previously Presented)** A method according to claim 48 wherein different ones of the compression routine control signals indicate a same compressions routine for different first threads.

51. **(Previously Presented)** A method according to claim 47, wherein different ones of the first threads include blocks of data containing different types of data.

52. **(Previously Presented)** A method according to claim 1, wherein different ones of the first threads include blocks of data containing different types of data.

53. **(Previously Presented)** A method according to claim 7, wherein different ones of the first threads include blocks of data that do not share common compression characteristics.

54. **(Previously Presented)** A method according to claim 19 wherein at least certain ones of the first threads are independently operated upon in parallel.

55. **(Previously Presented)** A method according to claim 4 wherein each first thread further includes control signals.

56. **(Previously Presented)** A method according to claim 55 wherein the control signals in each first thread include a compression routine control signal indicating a compression routine to be used in the step of operating.

57. **(Previously Presented)** A method according to claim 56 wherein different ones of the compression routine control signals indicate different compressions routines for different first threads.

58. **(Previously Presented)** A method according to claim 56 wherein different ones of the compression routine control signals indicate a same compressions routine for different first threads.

59. **(Previously Presented)** A method according to claim 55, wherein different ones of the first threads include blocks of data containing different types of data.

60. **(Previously Presented)** A method according to claim 59 wherein, during the step of operating upon each of the plurality of first threads, at least two different compression algorithms are used to independently operate upon different first threads.

61. **(Previously Presented)** A method according to claim 39, wherein the step of compressing the digital data using multiple passes includes the step of

partitioning the digital data into a plurality of blocks;

creating a plurality of first threads, such that each first thread includes at least one of the plurality of blocks; and

operating upon each of the plurality of first threads to obtain a plurality of compressed first threads, each compressed first thread including at least one compressed block of digital data.

62. **(Previously Presented)** A method according to claim 61 wherein at least certain ones of the first threads are independently operated upon in parallel.

63. **(Previously Presented)** A method according to claim 62 wherein, during the step of operating, at least two different compression algorithms are used to independently operate upon different first threads.

64. **(Previously Presented)** A method according to claim 61, wherein the step of compressing further includes the steps of:

operating upon each of the compressed first threads to eliminate each of the compressed first threads and retain the compressed first blocks;

creating a plurality of second threads, such that each second thread includes at least one of the plurality of compressed first blocks; and

operating upon each of the plurality of second threads to obtain a plurality of compressed second threads, each compressed second thread including at least one compressed second block of digital data.

65. **(Previously Presented)** A method according to claim 64 wherein at least certain ones of the second threads are independently operated upon in parallel.

66. **(Previously Presented)** A method according to claim 64 wherein, during the step of operating upon each of the plurality of second threads, the same compression algorithm used to operate upon each block is also used to operate upon the corresponding compressed block.

67. **(Previously Presented)** A method according to claim 61 wherein each first thread has an associated first metadata set.

68. **(Previously Presented)** A method according to claim on 67 wherein each first metadata set includes a passes required variable.

69. **(Previously Presented)** A method according to claim 61 wherein each first thread further includes control signals.

70. **(Previously Presented)** A method according to claim 69 wherein the control signals in each first thread include a compression routine control signal indicating a compression routine to be used in the step of operating.

71. **(Previously Presented)** A method according to claim 70 wherein different ones of the compression routine control signals indicate different compressions routines for different first threads.

72. **(Previously Presented)** A method according to claim 70 wherein different ones of the compression routine control signals indicate a same compressions routine for different first threads.

73. **(Previously Presented)** A method according to claim 61, wherein different ones of the first threads include blocks of data containing different types of data.

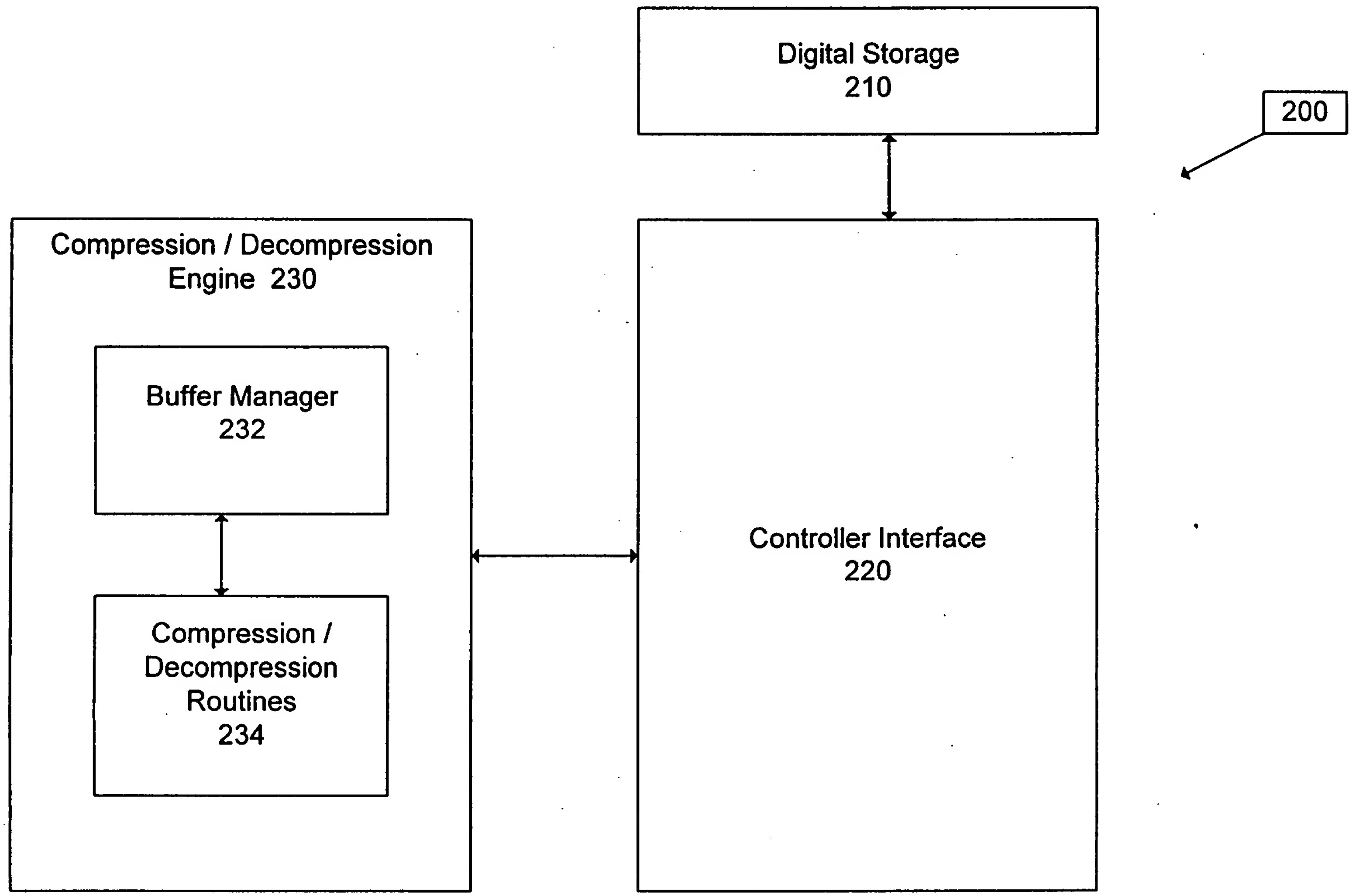
Digital Storage
210
200
Compression / Decompression
Engine 230
Buffer Manager
232
Compression /
Decompression
Routines
234
Controller Interface
220

Fig. 2